2013 21st Mediterranean Conference on Control & Automation (MED)
Platanias-Chania, Crete, Greece, June 25-28, 2013

# Optimizing Mixtures of Dependency Trees with Application to Distributed Probabilistic Control

Miguel Barão[1]

***Abstract*— One of the problems in distributed control is that of establishing a communication network topology between the intervening controllers that best suits the closed loop performance of the whole system. In this paper, a particular view of this problem is analyzed where the optimal actuation is described probabilistically and assumed to be jointly specified. The main problem is that of finding a topology having pairwise communication links that best approaches a joint distribution of actions at each time instant. The proposed algorithm uses properties of the natural gradient in the manifold of categorical distributions to find a mixture of dependency trees under certain network topology constraints.**

## I. Introduction

One central problem in distributed control is to make controllers act so that, collectively, they attain a certain desired global behavior. Due to the computational complexity and memory limitations of the agents, or limitations in the underlying communication network, the optimal theoretical behavior often can not be achieved.

One frequently studied scenario in the literature is the distributed model predictive control problem, where several model predictive controllers perform local optimizations while communicating their findings to their neighbors. In this framework, it is commonly assumed that the network supports several variable sharing iterations among the controllers at each time step in order to synchronize the controllers actions to a common good.

A different scenario occurs when we consider agents behaving stochastically. In this setting, controllers or agent actions are described probabilistically given the system state (see a formulation of probabilistic control in [6]). In this setting, a probabilistic control law is specified by a conditional distribution $p(u|x)$ that is obtained by a suitable optimization process. Assuming that a set of $n$ controllers actuate simultaneously on the same system, their joint action can be defined as a new variable $\mathbf{u} \triangleq (u_1, \ldots, u_n)$ and, at least conceptually, one could now solve the control optimization to find the collective behavior $p(\mathbf{u}|x) = p(u_1, \ldots, u_n|x)$. Despite its probabilistic caracterization, this solution is still centralized. If a distributed version is sought, each individual controller should instead be optimized allowing some sharing of information between them in order to approximate the optimal centralized solution. If the communication network is itself subject to constraints, the optimal centralized solution may not be achievable.

In this paper, each controller is allowed to receive a single packet of information at each time step, being the communicated information the action of another agent. The problem that now arises is the selection of which other controller should each one listen to such that the collective behavior is as close as possible to the optimal centralized solution. In order to avoid waiting deadlocks, the communication topology is shaped as a tree (to avoid communication loops) or a forest if several independent roots can be found.

The tree shape problem has been dealt with in the literature following the celebrated Chow-Liu algorithm [4]. This algorithm finds the best dependency tree that approximates a given joint distribution or, alternatively, learns it from data using the maximum likelihood criteria.

A dependency tree can be used to define a fixed network topology. Albeit its use is appropriate if the network is constrained to have a fixed topology, for networks allowing dynamic links, then a time varying topology can potentially lead to better results. This leads to a mixture of trees where each agent can select another agent to listen to, from a set of available agents according to some mixing probabilities.

Mixture of trees were introduced in [10], [9] where the mixture coefficients are learned from data via the expectation-maximization algorithm (EM) following either a maximum likelihood (ML) or a Bayesian/maximum *a posteriori* (MAP) criteria. The current work follows a different path, where it is assumed that the target distribution is a joint probabilistic controller $p(\mathbf{u}|x)$.

The main contributions of the paper are the formulation of the dynamic topology as a mixture of dependency trees, and the advantage taken from the use of the natural gradient to enforce constraints simultaneously on probabilities and on the network topology.

The paper is organized as follows: section II describes dependency trees and the Chow-Liu algorithm; section III motivates the use of mixture of dependency trees and formulates the problem to be solved; section IV introduces the natural gradient in the probability manifold and proves some useful properties that are used to impose constraints in the network topology; finally section VI draws conclusions.

## II. Problem formulation

It is assumed that a collection of $n$ controllers generate a collective actuation signal $(u_1, \ldots, u_n)$ depending on the current system state, which is assumed to be known. This collective action can be written, at each time instant, as a

This work was supported by national funds through FCT – Fundação para a Ciência e a Tecnologia, under project PEst-OE/EEI/LA0021/2011 and project PROBCONTROL - PTDC/EEA-CRO/115038/2009.

[1]M. Barão is with the Informatics Department, University of Évora, and Control of Dynamical Systems Group, INESC-ID Lisboa.

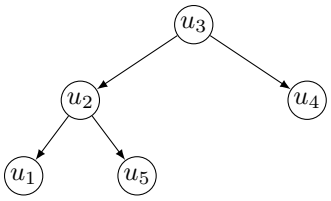


Fig. 1. Dependency tree used to approximate a joint distribution $p(u_1, \ldots, u_6)$.

conditional probability distribution $p(u_1, \ldots, u_n|x)$, where $x$ is the system state. Knowing the desired closed loop behavior of the system, it is possible, in principle, to derive an optimal collective action $p(u_1, \ldots, u_n|x)$. We refer to [6], [11] for such a probabilistic formulation.

In general, the actions $(u_1, \ldots, u_n)$ are dependent variables. This dependency means that, if they are to be implemented separately, they have to communicate with each other. The dependency can be made explicit using the chain rule of probabilities

$$\begin{aligned} p(u_1, \ldots, u_n|x) = \\ = p(u_1|x)p(u_2|u_1, x) \cdots p(u_n|u_1, \ldots, u_{n-1}, x). \end{aligned} \tag{1}$$

According to this particular factorization, action $u_1$ is generated independently, then action $u_2$ is generated depending on the value $u_1$ communicated by the first agent, then action $u_3$ is generated according the the two preceding ones, and so forth. In this communication topology, the complexity required to represent each control law grows exponentially with the number of agents. For instance, if each agent has a finite set $\mathcal{A}$ of possible actions, then the $n$th agent's behavior $p(u_n|u_1, \ldots, u_{n-1}, x)$ is represented by a table with $\#\mathcal{A}^n$ entries for each state $x$.

To have a realizable implementation, a naive possibility is to consider independence between agents, *i.e.* no communication. In this case, each one acts independently according to a marginal distribution $p(u_i|x)$. These scheme, however can produce results far from the optimal collective behavior if coordination is required. The next level of complexity is to consider pairwise dependencies between controllers.

### A. Pairwise dependencies

Using this approach, each controller's action depends on a single communicated variable from a different controller, *i.e.*, the control law for the controller $i$ is described by a conditional distribution $p(u_i|u_j, x)$, where $u_j$ is the communicated action from controller $j$. Under this assumption, the communication topology is required to be a directed acyclic graph like the one depicted in figure 1. The joint distribution obtained from the dependency tree of figure 1 is

$$\begin{aligned} p_a(u_1, \ldots, u_5) = \\ = p(u_1|u_2)p(u_2|u_3)p(u_3)p(u_4|u_3)p(u_5|u_2). \end{aligned} \tag{2}$$

For notational convenience in the current section, the state $x$ is dropped to simplify the notation, although the controllers actions depend on it. Generally, for $n$ agents, the obtained joint probability distribution is

$$p_a(u_1, \ldots, u_n) = \prod_{i=1}^{n} p(u_i|u_{j(i)}), \tag{3}$$

where $j(i)$ is the ancestor of node $i$ in the dependency tree.

Many such graphs are possible alternatives, each one rendering a different joint distribution $p_a(u_1, \ldots, u_n)$. These approximated distributions are generally different from the optimal one, $p^*(u_1, \ldots, u_n)$.

A problem that know can be posed is that of finding the dependency tree such that its joint distribution $p_a(u_1, \ldots, u_n)$ is the closest one to the target distribution $p^*(u_1, \ldots, u_n)$. This is precisely the problem solved in [4].

### B. Brief description of the Chow-Liu algorithm

In this algorithm the Kullback-Leibler divergence is used to assess the quality of the approximation. The Kullback-Leibler divergence is a nonnegative function that is zero if and only if $p^* = p_a$. It is defined by

$$D(p^* \| p_a) \triangleq \sum_{u_1, \ldots, u_n} p^*(u_1, \ldots, u_n) \log \frac{p^*(u_1, \ldots, u_n)}{p_a(u_1, \ldots, u_n)}. \tag{4}$$

It can be roughly though as a distance measure between probability distributions but it's not strictly a distance since it's not symmetric and does not satisfy the triangular inequality.

The Chow-Liu algorithm then finds the tree $j(i)$ that minimizes the Kullback-Leibler divergence (4) between the true distribution $p*$ and its approximation $p_a$:

$$\min_{j(i)} D(p^* \| p_a). \tag{5}$$

It can be shown that the divergence $D(p^* \| p_a)$ can be written as

$$D(p^* \| p_a) = \sum_{i=1}^{n} H(u_i) - H(u_1, \ldots, u_n) - \sum_{i=1}^{n} I(u_i; u_{j(i)}), \tag{6}$$

where $H(\cdot)$ is the entropy function defined by

$$H(u_1, \ldots, u_n) = - \sum_{u_1, \ldots, u_n} p(u_1, \ldots, u_n) \log p(u_1, \ldots, u_n) \tag{7}$$

and $I(u_i; u_{j(i)})$ is the mutual information between variables $u_i$ and $u_{j(i)}$ defined by

$$I(u_i; u_{j(i)}) \triangleq D\Big(p(u_i, u_{j(i)}) \Big\| p(u_i)p(u_{j(i)})\Big). \tag{8}$$

Since the entropy terms do not depend on the tree, minimizing (6) with respect to $j(i)$ amounts to maximize the sum of the mutual information corresponding to the edges of the tree. The Chow-Liu algorithm then proceeds to find the tree maximizing this sum using the Kruskal algorithm [8], a well known algorithm to find minimum spanning trees (see also [5]).

## III. Mixing dependency trees

The previous section shown that the Chow-Liu algorithm can be used to find a communication topology so that controllers under a pairwise dependency constraint approximate the optimal control $p^*(u_1,\dots,u_n|x)$. The solution found fixes a particular network topology with rigid communication links that is then used every time the process is run.

If the communication network is not restricted to have rigid links, each controller selects which one is to be observed at each time instant. We shall assume that this selection is described probabilistically so that agent $i$ observes the action $u_j$ from agent $j$ with probability $p_i(j)$. Then, the probability distribution for action $u_i$ from agent $i$ is the mixture

$$p(u_i)p_i(i) + \sum_{\substack{j=1\\ j\neq i}}^{n} p(u_i|u_j)p_i(j), \tag{9}$$

where the left term represents an action taken independently without communication. Since the probabilities add up to one, we can write $p_i(i) = 1 - \sum_{j\neq i} p_i(j)$ and replace it in (9) to yield the mixture

$$p(u_i) + \sum_{\substack{j=1\\ j\neq i}}^{n} \big(p(u_i|u_j) - p(u_i)\big)p_i(j), \tag{10}$$

where only the $n-1$ probabilities $p_i(j)$ with $j\neq i$ are now taken into account.

The joint distribution resulting from this mixture is given by the product

$$p_a(u_1,\dots,u_n) = \\ = \prod_{i=1}^{n}\left(p(u_i) + \sum_{\substack{j=1\\ j\neq i}}^{n}\big(p(u_i|u_j) - p(u_i)\big)\,p_i(j)\right). \tag{11}$$

It can be checked that this model includes the earlier fixed dependency tree as a particular case by forcing $p_i(j)$ to be a Kronecker delta function centered at $j(i)$ or identically zero if no communication occurs (root nodes). The problem that has to be solved in the mixing trees realm is that of finding the distribution $p_i(j)$ for each controller $i = 1,\dots,n$.

Unfortunately, if all $p_i(j)$ are allowed to be positive, a sampling of this distribution will likely produce a network topology with dependency loops leading to communication deadlocks. Figure 2 exemplifies a deadlock situation that can arise when all communication links are available.

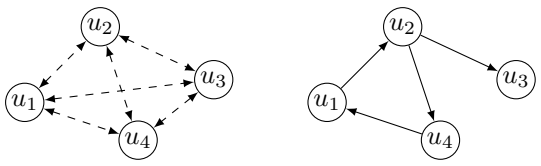


Fig. 2. Unconstrained topology (left) and communication deadlock (right).

To avoid these situations, the allowed topologies are encoded in $p_i(j)$ and are constrained into a lower triangular matrix

$$\begin{bmatrix} p_1(1) & 0 & \cdots & 0 \\ p_2(1) & p_2(2) & \ddots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ p_n(1) & p_n(2) & \cdots & p_n(n) \end{bmatrix} \tag{12}$$

where each line specifies a probability distribution. According to this arrangement, controller $i$ can only observe $j<i$ or generate its action independently (Note that the diagonal elements are not independent parameters. Instead they are determined so that probabilities add one).

To optimize the mixture parameters $p_i(j)$, the Kullback-Leibler divergence (4) is used to assess the quality of the approximation. Expanding (4) yields

$$D(p^*\|p_a) = -H(p^*) - \sum_{i=1}^{n}\sum_{u_1,\dots,u_n} p^*(u_1,\dots,u_n)\cdot \\ \cdot \log\left(p(u_i) + \sum_{j<i}\big(p(u_i|u_j) - p(u_i)\big)p_i(j)\right), \tag{13}$$

where the first term is the entropy of the distribution $p^*$. This term does not depend on the parameters $p_i(j)$ and can be discarded on an optimization procedure. The second term can be interpreted as the sum of the expected Kerridge inaccuracies [7]

$$K\big(p^*(u_i|\mathbf{u}_{-i}) : p_a\big) \triangleq -\sum_{u_i} p^*(u_i|\mathbf{u}_{-i})\log p_a(u_i|\mathbf{u}_{-i}) \tag{14}$$

and is the function to be optimized. The notation $\mathbf{u}_{-i}$ is used here to indicate all variables $u_1,\dots,u_n$ excluding $u_i$.

To find a minimum of (13), we differentiate with respect to the mixture probabilities $p_i(j)$, relabeled here to $p_r(s)$ to avoid clashing with the summation indices. The stationarity points are given by

$$0 = \frac{\partial}{\partial p_r(s)} D(p^*\|p_a) = \sum_{u_1,\dots,u_n} p^*(u_1,\dots,u_n)\cdot \\ \cdot \frac{p(u_r|u_s) - p(u_r)}{p(u_r) + \sum_{j<r}\big(p(u_r|u_j) - p(u_r)\big)p_r(j)}, \tag{15}$$

for $r = 1,\dots,n$ and $s < r$. Since this equation does not seem to have a simple explicit solution, an iterative method is employed to solve it. Applying directly the gradient method has the drawback that we are dealing with probability constraints, namely they are nonnegative, add up to one, and some of them can be constrained to zero to avoid communication deadlocks. To deal with all this constraints, a natural gradient method is used on the manifold of categorical probability distributions. It is proven next that this method deals with all of the above constraints implicitly and can therefore be implemented as an unconstrained optimization algorithm. The next section describes this method

and proves some of it's properties that are relevant in the current problem.

## IV. The natural gradient method

The gradient vector of an arbitrary function $f$ is defined as the vector $\nabla f$ that satisfies the equation

$$\langle \nabla f, \mathbf{v} \rangle = \mathbf{d}f(\mathbf{v}), \quad \forall \mathbf{v} \neq \mathbf{0}. \tag{16}$$

In this equation, $\langle \cdot, \cdot \rangle$ denotes an inner product and $\mathbf{d}f$ is the differential, or one-form, of $f$ (see [3] for an introduction on differential geometry and the gradient vector).

For a particular function $f$, the gradient vector $\nabla f$ depends on the specific metric considered. When dealing with probability manifolds it has long been suggested [1], [2] that a natural inner product is obtained by

$$\langle \mathbf{v}, \mathbf{w} \rangle \triangleq \sum_{i,j} v^i w^j g_{ij}, \tag{17}$$

where the $v^i, w^j$ are the components of arbitrary vectors $\mathbf{v}$ and $\mathbf{w}$, and the metric tensor $g_{ij}$ is given by the Fisher information matrix $\mathbf{G} \triangleq [g_{ij}]$, and computed by

$$g_{ij} \triangleq E_\theta \left[ \frac{\partial \log p}{\partial \theta^i} \frac{\partial \log p}{\partial \theta^j} \right]. \tag{18}$$

The symbols $\theta^i$ denote parameters of the probability distribution $p$. Specializing for a categorical probability distribution in column form

$$\mathbf{p} = \begin{bmatrix} p_1 \\ \vdots \\ p_i \\ \vdots \\ p_n \end{bmatrix} = \begin{bmatrix} p_1 \\ \vdots \\ 1 - \sum_{j \neq i} p_j \\ \vdots \\ p_n \end{bmatrix} \tag{19}$$

having $n-1$ independent parameters $p_j$, $j \neq i$, the Fisher information matrix (18) is given by

$$\mathbf{G} = \frac{1}{1 - \sum_{j \neq i} p_j} \begin{bmatrix} 1 & \cdots & 1 \\ \vdots & & \vdots \\ 1 & \cdots & 1 \end{bmatrix} + \begin{bmatrix} p_1 & & \underline{0} \\ & \ddots & \\ \underline{0} & & p_n \end{bmatrix}^{-1}, \tag{20}$$

where the matrices are $(n-1) \times (n-1)$ and the probability $p_i$ is missing from the diagonal of the second matrix.

The use of the Fisher information matrix $\mathbf{G}$ to define the inner product in (16) leads to a linear system of equations that in matrix form reads

$$\begin{aligned} (\nabla f)^T \mathbf{G} \mathbf{v} &= \mathbf{d}f(\mathbf{v}) \\ &= \begin{bmatrix} \frac{\partial f}{\partial p_1} & \cdots & \frac{\partial f}{\partial p_{i-1}} & \frac{\partial f}{\partial p_{i+1}} & \cdots & \frac{\partial f}{\partial p_n} \end{bmatrix} \mathbf{v}, \end{aligned} \tag{21}$$

for all $\mathbf{v} \neq \mathbf{0}$. Its solution is given by

$$\nabla f = \mathbf{G}^{-1} \begin{bmatrix} \frac{\partial f}{\partial p_1} & \cdots & \frac{\partial f}{\partial p_{i-1}} & \frac{\partial f}{\partial p_{i+1}} & \cdots & \frac{\partial f}{\partial p_n} \end{bmatrix}^T. \tag{22}$$

Using the matrix inversion lemma, the Fisher information matrix (20) can be inverted and the gradient becomes

$$\nabla f = \left( \begin{bmatrix} p_1 & & \underline{0} \\ & \ddots & \\ \underline{0} & & p_n \end{bmatrix} - \begin{bmatrix} p_1 \\ \vdots \\ p_n \end{bmatrix} \begin{bmatrix} p_1 & \cdots & p_n \end{bmatrix} \right) \begin{bmatrix} \frac{\partial f}{\partial p_1} \\ \vdots \\ \frac{\partial f}{\partial p_n} \end{bmatrix}, \tag{23}$$

again omitting probability $p_i$ from the matrices. Distributing the Jacobian matrix on the right simplifies further the equation so that the gradient can be computed with linear computational complexity and without explicitly computing the Fisher information matrix.

The natural gradient just presented has several useful properties that can be exploited in the problem studied in this paper.

*Propostition 1:* The gradient flow realized as the dynamical system

$$\dot{\mathbf{p}} = -\nabla f \tag{24}$$

satisfies, intrinsically, the probability constraints

$$p_j \geq 0, \qquad \sum_{j=1}^{n} p_j = 1. \tag{25}$$

*Proof:* The equilibrium points of (24) are the solutions to the equation

$$\mathbf{0} = \mathbf{G}^{-1} (\mathbf{d}f)^T. \tag{26}$$

The solution of this equation is composed by the zeros of differential $\mathbf{d}f = \mathbf{0}$, plus the points where $\mathbf{d}f$ is in the kernel of $\mathbf{G}^{-1}$.

When all probabilities $p_j$ are positive, it can be checked that $\mathbf{G}^{-1}$ is nonsigular and thus, the possible solutions are the same as the ones obtained with the usual Euclidean gradient.

If one probability $p_k$ becomes zero, the matrix $\mathbf{G}^{-1}$ becomes singular and its null space cancels any deviation of $p_k$ from zero. This cancelation is progressive as the probability $p_k$ approaches zero. To see this, we factorize $\mathbf{p} = \mathbf{A}\sqrt{\mathbf{p}}$ where

$$\mathbf{A} \triangleq \begin{bmatrix} \sqrt{p_1} & & \underline{0} \\ & \ddots & \\ \underline{0} & & \sqrt{p_n} \end{bmatrix}, \qquad \sqrt{\mathbf{p}} \triangleq \begin{bmatrix} \sqrt{p_1} \\ \vdots \\ \sqrt{p_n} \end{bmatrix}. \tag{27}$$

Then the following equalities hold

$$\begin{aligned} \mathbf{G}^{-1} &= \mathbf{A}\mathbf{A}^T - \mathbf{A}\sqrt{\mathbf{p}}\sqrt{\mathbf{p}}^T\mathbf{A}^T \\ &= \mathbf{A}(\mathbf{I} - \sqrt{\mathbf{p}}\sqrt{\mathbf{p}}^T)\mathbf{A}^T \\ &= \mathbf{A}\left( \mathbf{I} - \|\sqrt{\mathbf{p}}\|^2 \frac{\sqrt{\mathbf{p}}}{\|\sqrt{\mathbf{p}}\|} \frac{\sqrt{\mathbf{p}}^T}{\|\sqrt{\mathbf{p}}\|} \right) \mathbf{A}^T, \end{aligned} \tag{28}$$

where the squared Euclidean norm is

$$\|\sqrt{\mathbf{p}}\|^2 = \sum_{j \neq i} p_j = 1 - p_i \tag{29}$$

and the vectors $\sqrt{\mathbf{p}}/\|\sqrt{\mathbf{p}}\|$ are unitary vectors with respect to the Euclidean norm.

Finally, we define the matrices

$$\mathbf{H}_i \triangleq \mathbf{I} - (1-p_i)\frac{\sqrt{\mathbf{p}}}{\|\sqrt{\mathbf{p}}\|}\frac{\sqrt{\mathbf{p}}^T}{\|\sqrt{\mathbf{p}}\|} \tag{30}$$

$$\mathbf{H}_j \triangleq \mathbf{I} - (1-p_j)\mathbf{e}_j\mathbf{e}_j^T, \quad j \neq i, \tag{31}$$

where $\mathbf{e}_j$ denote the standard basis vectors. This matrices perform projections when $p_i = 0$ and $p_j = 0$, respectively. Otherwise attenuations are performed along the directions $\sqrt{\mathbf{p}}$ and $\mathbf{e}_j$. Then $\mathbf{G}^{-1}$ can be written as the product

$$\begin{aligned}\mathbf{G}^{-1} &= \mathbf{A}\mathbf{H}_i\mathbf{A}^T \\ &= \mathbf{A}\mathbf{H}_i\mathbf{A}^{-1}\mathbf{A}\mathbf{A}^T, \qquad p_j > 0 \\ &= \mathbf{A}\mathbf{H}_i\mathbf{A}^{-1}\prod_{j\neq i}\mathbf{H}_j.\end{aligned} \tag{32}$$

Using this factorization we see that $\mathbf{G}^{-1}(\mathbf{d}f)^T$ performs a sequence of scalings $\mathbf{H}_j$ of the $j$-th component of the differential $\mathbf{d}f$. This scalings tend to cancel the components of the differential $\mathbf{d}f$ that correspond to probabilities close to zero. They perform an exact cancelation when a given probability $p_k$ is exactly zero since in this case $\mathbf{H}_k$ becomes a projection

$$\mathbf{H}_k = \mathbf{I} - \mathbf{e}_k\mathbf{e}_k^T, \tag{33}$$

into the subspace orthogonal to $\mathbf{e}_k$. The factor $\mathbf{A}\mathbf{H}_i\mathbf{A}^{-1}$ works similarly but for the constraint $p_i = 1 - \sum_{j\neq i} p_j > 0$.

In this case, we prove that if $p_i = 0$ then it becomes stuck at zero, *i.e.* $\dot{p}_i = 0$. Let $\xi \triangleq \big(\prod_{j\neq i}\mathbf{H}_j\big)(\mathbf{d}f)^T$ denote the factors on the right of the gradient descent equation colapsed to a single variable $\xi$. Then

$$\begin{aligned}\dot{p}_i &= -\sum_{j\neq i}\dot{p}_j \\ &= -\sum_{j\neq i}\mathbf{A}\mathbf{H}_i\mathbf{A}^{-1}\Big(\prod_{j\neq i}\mathbf{H}_j\Big)(\mathbf{d}f)^T \\ &= -\begin{bmatrix}1 & \cdots & 1\end{bmatrix}\mathbf{A}\mathbf{H}_i\mathbf{A}^{-1}\xi \\ &= -\begin{bmatrix}1 & \cdots & 1\end{bmatrix}\mathbf{A}(\mathbf{I} - \sqrt{\mathbf{p}}\sqrt{\mathbf{p}}^T)\mathbf{A}^{-1}\xi \\ &= -\begin{bmatrix}1 & \cdots & 1\end{bmatrix}(\mathbf{I} - \mathbf{p}\begin{bmatrix}1 & \cdots & 1\end{bmatrix})\xi \\ &= -\begin{bmatrix}1 & \cdots & 1\end{bmatrix}\xi - \sum_{j\neq i}p_j\begin{bmatrix}1 & \cdots & 1\end{bmatrix}\xi \\ &= -\underbrace{(1 - \sum_{j\neq i}p_j)}_{=p_i=0}\begin{bmatrix}1 & \cdots & 1\end{bmatrix}\xi \\ &= 0.\end{aligned} \tag{34}$$

We then conclude that the admissible region for the probabilities is bounded by an "equilibria" submanifold of (24). At the boundary, the differential $\mathbf{d}f$ is projected to the submanifold and does not allow probability constraints to be violated. Furthermore, if the optimization starts with any given null probabilities $p_k = 0$, these will be kept canceled out while running the steepest descent even when $\mathbf{d}f$ points inward into the admissible region. ■

## V. Optimization of the mixture of trees

Returning to the problem of finding the mixture parameters $p_i(j)$, it can be seen that the properties proved in the previous section can be used to impose the communication topology constraints at the same time that the usual probability constraints are met. This is accomplished by a simple initialization of the mixture probabilities $p_i(j)$ to zero, as in (12), whenever no communication link is allowed.

The admissible trees, encoded in the triangular matrix (12), lie on a probability simplex since the valid matrices can be written as convex combinations of basis matrices. Since this probability simplex is a convex set and since the sum of expected Kerridge inaccuracies is convex, the solution found is guaranteed to be global.

## VI. Conclusion

This paper dealt with the estimation of a mixture of dependency trees in order to approximate some desired collective behavior of controller actions specified in a joint distribution $p^*(u_1, \ldots, u_n|x)$. It is assumed that the network topology is randomized from a set of admissible trees. The problem formulated here is that of estimating the mixture coefficients. To solve this problem a natural gradient algorithm is employed that takes advantage of its intrinsic properties in order to enforce constraints on the network topology without additional effort. The relevant properties are shown to hold in categorical distributions as is the case of the problem considered.

There are still open issues to be solved, namely the fact that the algorithm requires the admissible topologies to be provided from the start.

## References


[1] S.-I. Amari. *Differential-Geometrical Methods in Statistics*. Lecture Notes in Statistics. Springer-Verlag, 1985.

[2] S.-I. Amari and H. Nagaoka. *Methods of Information Geometry*. AMS, 2000.

[3] W. M. Boothby. *Differentiable Manifolds and Riemannian Geometry*. Academic Press, 1986.

[4] C. K. Chow and C. N. Liu. Approximating discrete probability distributions with dependence trees. *IEEE Transactions on Information Theory*, 14:462–467, 1968.

[5] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein. *Introduction to Algorithms*. MIT Press, 2001.

[6] M. Kárný. Towards fully probabilistic control. *Automatica*, 32(12):1719–1722, 1996.

[7] D. F. Kerridge. Inaccuracy and inference. *Journal of the Royal Statistical Society. Series B (Methodological)*, 23(1):184–194, 1961.

[8] J. B. Kruskal. On the shortest spanning subtree of a graph and the traveling salesman problem. In *Proceedings of the American Mathematical Society*, volume 7:1, pages 48–50, Feb 1956.

[9] M. Meila and M. Jordan. Learning with mixtures of trees. *Journal of Machine Learning Research*, 1:1–48, 2000.

[10] M. Meila and M. I. Jordan. Estimating dependency structure as a hidden variable. In M. I. Jordan, M. J. Kearns, and S. A. Solla, editors, *Neural Information Processing Systems*, volume 10, pages 584–590. MIT Press, 1998.

[11] E. Nováková and M. Kárný. Fully probabilistic control design for Markov chains. In *European Control Conference*, 1997.